

**Amendments to the Specification:**

**Please delete the paragraph beginning on page 4, line 1 and replace with the following:**

Referring now to the drawings in detail, wherein like reference numerals indicate like elements throughout, Figure 1 illustrates a computer system generally designated 10 according to the present invention. System 10 comprises a client 12, a server computer 14 and an intervening, known communication system 16. Client 12 can be a desktop or lap top personal computer, PDA, cell phone, electronic instrument/tool, equipment, etc. In the illustrated embodiment, client 12 includes a CPU 20, an operating system 22 executing on the CPU 20, and an application 24 executing on the operating system and CPU. Application 24 can be a browser, e-mail agent or connectivity application. Server 14 comprises one or more CPUs 30, an operating system 32 executing on CPU 30, and server software 34 executing on operating system 32 and CPU 30. Server software 34 can be an e-mail service web site or connectivity software. The communication system 16 comprises components such as electrical or optical transceivers and associated electrical or optical cabling, wireless communication transceivers, and/or networking hardware. Various, known communication protocols can be used for the communication between the client 12 and the server 14 such as SOAP, TCP/IP, HTTP, HTML, FTP, or SMTP.

**Please delete the paragraph beginning page 12, line 25 and replace with the following:**

Referring again to decision 111, if the current message includes a compressed header, i.e. a UID and changes, if any, to the previously cached header referenced by the UID, then server software 34 recreates the full header (step 120). The server software 34 recreates the full header by reading from cache 56 the previously cached header referenced by the UID in the current compressed message header, and then modifying the previously cached header with the changes, if any, in the current message header of the compressed message. Next, the server software combines the recreated, full header with the current payload to form a full, current message (step 122). Then, server software assigns a new UID for the recreated, full current header and caches the recreated, full current header (step 124). Then, server software 34 handles and responds to the recreated, full current message in the prior art manner, except that the server software 34 will include the UID in the response (step 115). The UID represents the recreated, full current header (if a compressed header was furnished by the client in step 100) or the UID for the full current header (if the full header was furnished by the client in step 100) ~~(step 130)~~. The server

software 34 also returns an acknowledgment that the message was handled. For subsequent messages, the client 12 can opt to use this UID (or other UID previously or subsequently returned by server 14) to compress the respective message from the client. Compression of the messages will reduce the bandwidth/transmission time requirement for the messages sent from the client to the server. As noted above, if the server software 34 is not willing to support header compression for this type or any type of message from client 12, then the server software 34 does not return a UID in step 110 [[130]], and instead returns a simple acknowledgment that the message was handled.